CLAIMS

What is claimed is:

1. A method for linking a first software module with a second software module comprising the steps of:

storing a public key in said first software module;

associating a stub digitally signed by an owner of said public key with said second software module;

computing a hash function value on a portion of said second software module; and

linking said first software module with said second software module upon verifying by use of said public key said digital signature on said stub and that saidcomputed hash function value equals a hash function value included in said digitally signed stub.

2. The method of claim 1 wherein said second software module is one of a plurality of software modules to be linked and said first software module includes a plurality of previously linked software modules.

3. The method of claim 1 wherein the steps of computing and verifying are performed by a dedicated processor.

4. A method for linking a first software module with a second software module comprising the steps of:

storing a first hash function value in said first software module;

computing a second hash function value on a portion of said contents of said second software module; and linking said first software module with said second software module upon verifying that said second hash function value is equal to said first hash function value.

5. The method of claim 4 wherein said second software module is one of a plurality of software modules to be linked and wherein said first software module includes a plurality of previously linked software modules.

6. The method of claim 4 wherein the steps of computing and verifying are performed by a dedicated processor.

7. A user device comprising:

a first storage module; and

a second storage module into which a software module is stored, verification software stored in said first storage module which verifies that a portion of said software module is authorized by computing a hash function value on said portion and comparing said computed hash function value with a hash function value stored in said verification software.

8. The user device of claim 7 wherein said first storage module is difficult to modify by software means.

9. The user device of claim 7 further comprising:

a public key within said first storage module;

an additional verification software within said first storage module; and

a digitally signed stub within said second storage module associated with said second software module, said additional verification software computes a hash function value on a portion of said second software module and verifies by

09750274 .110300

use of said public key that said digitally signed stub includes a digital signature on said computed hash function value.

10. A user device comprising:

a first storage module having a public key and verification software stored therein; and

a second storage module into which a software module is stored, a digitally signed stub associated with said second software module, said verification software computes a hash function value on a portion of said second software module and verifies by use of said public key that said digitally signed stub associated with said second software module includes a digital signature on said computed hash function value.

11. The user device of claim 10 further comprising:

a hash function value within said first storage module wherein said verification software further verifies that a portion of said second software module is authorized by computing a hash function value on said portion and comparing said computed hash function value with a hash function value stored in said verification software.

12. A watchdog program comprising:

a value;

a function to compute; and

means for verifying a software module stored in memory by computing said function on a sequence of locations in said software module and comparing said result of said computation with said value.

09705074.110300

13. The watchdog program of claim 12 wherein said function is a hash function and said means for verifying computes said hash function value on said sequence of locations and compares said result of said computation with said value.

14. The watchdog program of claim 12 further comprising:

a watchdog action; and

means for performing said watchdog action dependent on said result of said comparison.

15. The watchdog program of claim 14 wherein said watchdog action includes halting said operation of a user device on which said watchdog program is executing.

09705074.110300

16. The watchdog program of claim 12 further comprising:

a plurality of watchdog actions; and

means for performing at least one watchdog action dependent on said result of said comparison.

17. The watchdog program of claim 12 further comprising:

means for performing a need-to-check test; and

means for determining whether to perform said function on said sequence of locations in said software module dependent on said result of said need-to-check test.

18. The watchdog program of claim 12 further comprising:

a plurality of sequences of locations; and

a plurality of stored values.

19. The watchdog program of claim 18 further comprising:

a plurality of watchdog actions; and

means for performing at least one said watchdog action dependent on said result of a comparison between values computed by said function on said plurality of sequences of locations and said stored values.

20. The watchdog program of claim 18 further comprising:

a plurality of memory locations;

means for selecting a software module; and

means for storing a start execution time and an end execution time for said software module in said memory locations.

21. The watchdog program of claim 12 wherein said watchdog program is a subroutine stored in a watchdog field in another program.

22. The watchdog program of claim 21 wherein said subroutine is placed within said watchdog field in a location dependent on conditions present when said another program is loaded.

23. The watchdog program of claim 21 wherein said location of said subroutine is changed after said subroutine is loaded.

24. The watchdog program of claim 21 wherein said placement of calls to said subroutine is determined based on conditions present when said another program is loaded.

25. The watchdog program of claim 21 wherein said placement of calls to said subroutine is changed after said subroutine is loaded.

26. The watchdog action of claim 14 wherein said watchdog action includes moving other watchdog subroutines within a watchdog field.

27. A tag table comprising:

a tag table identifier having a value;

a tag for a copy of software, said tag comprising said tag table identifier value and a hash function value of a portion of said copy of software; and

a digitally signed message, said digitally signed message comprising said tag table identifier value and said hash function value.

28. The tag table of claim 27 wherein said tag further comprises a usage policy and said digitally signed message further comprises a usage policy.

29. The tag table of claim 27 wherein said tag further comprises a name and said digitally signed message further comprises a name.

30. The tag table of claim 27 further comprising:

a hash function value for said tag table sent from a guardian center in a previous guardian center call-up.

31. The tag table of claim 27 further comprising:

a header, said header including a continuation message sent from a guardian center in a previous guardian center call-up.

32. The tag table of claim 27 further comprising:

usage statistics for said copy of software.

33. A method for purchasing software comprising the steps of:

09705074.110300

creating, by a purchaser, a data structure including a tag table identifier value associated with a tag table in a user device and an identification of said software;

computing, by said purchaser, a hash function value of said data structure;

sending, by said purchaser, a message to a vendor, said message comprising said hash function value and said identification of said software.

34. The method of claim 33 further comprising the steps of:

upon receiving said message, said vendor digitally signing said message and returning said signed message to said purchaser;

verifying by a supervising program on said user device said digital signature on said signed message by use of said vendor's public key; and

verifying by said supervising program that said signed message includes said message sent by said purchaser.

35. The method of claim 33 comprising the step of:

establishing a secure communication channel between said purchaser and said vendor before sending said message.

36. The method of claim 34 further comprising the steps of:

storing a hash function value of a portion of said software in said identification of software; and

verifying, by said supervising program, that said hash function value in said identification of said software equals a computed hash function value on said portion of said software.

37. The method of claim 34 further comprising the step of:

09705740.110300

storing, by said supervising program, a tag for said software in said tag table wherein said tag includes said tag table identifier value, said purchaser-created data structure, and said signed message.

38. The method of claim 33 wherein said data structure further includes a usage policy and said message further comprises said usage policy.

39. The method of claim 33 wherein said data structure further includes a new randomly chosen value occurring only once.

40. The method of claim 33 wherein said message further includes a proof of payment for said software.

41. A method for decommissioning a copy of software in a user device comprising the steps of:

removing, by a supervising program, a tag associated with said copy of software from a tag table in said user device, said tag including a digitally signed portion and a tag table identifier value;

establishing a communications channel from said user device to a vendor;

sending, by said user device said tag to said vendor on said communication channel;

verifying, by said vendor, said digital signature on said digitally signed portion by use of said public key of said vendor; and

reading, by said vendor, said tag table identifier value.

42. The method of claim 41 further comprising the step of:

sending by said vendor a certificate of credit to a purchaser of said tag.

09730474 .120300

43. The method of claim 41 further comprising the steps of:

sending, by said vendor said digitally signed portion and said tag table identifier value to a guardian center;

storing by said guardian center said digitally signed portion of said tag; and

linking, by said guardian center, said digitally signed portion of said tag to said tag table identifier value.

44. The method of claim 43 further comprising the step of:

transmitting, by said guardian center, a continuation message to said supervising program in said user device, said continuation message including said digitally signed portion of said tag and said tag table identifier value.

45. The method of 44 further comprising the step of:

verifying, by said supervising program that said digitally signed portion of said tag having said tag table identifier value is not stored in said tag table.

46. A method for supervising usage of software on a user device comprising the steps of:

computing, by a supervising program within said user device, a first hash function value of a tag table;

sending, by said supervising program, a call-up message to a guardian center, said call up message comprising said first hash function value, an identifier value of said tag table, and a second hash function value of said tag table sent in a previous call-up message;

verifying, by said guardian center, that said hash function value of said tag table sent in said previous call-up message is a most recently stored value in a list of hash function values stored by said guardian center and associated with said identifier value of said tag table;

09706074.110300

upon successful verification by said guardian center, appending said received first tag table hash function value to said list of hash function values associated with said identifier value of said tag table; and

sending, by said guardian center, a digitally signed continuation message to said supervising program, said continuation message comprising a portion of said call-up message.

47. The method of claim 46 further comprising the step of:

verifying, by said supervising program, that a portion of said digitally signed guardian center message is equal to said corresponding portion sent in said call-up message.

48. The method of claim 47 further comprising the step of:

upon failure of said verification, initiating by said supervising program a new call-up to said guardian center.

49. The method of claim 46 wherein said guardian center stores said received call-up message and said continuation message and associates said stored messages with said tag table identifier value.

50. The method of claim 49 further comprising the step of:

upon receiving a call-up message from said supervising program, sending, by said guardian center, said stored continuation message upon verifying that said received call-up message equals said stored call-up message.

51. The method of claim 46 wherein the step of verifying further comprises the step of:

09702074.110300

upon failure of said verification, sending, by said guardian center, a digitally signed message to said calling supervising program indicating said failure.

52. The method of claim 51 further comprising the step of:

upon receiving said digitally signed message from said guardian center, invalidating said tag table by said supervising program.

53. The method of claim 46 wherein the step of verifying further comprises the step of:

upon failure of said verification, rejecting, by said guardian center future call-ups including said tag table identifier value.

54. The method of claim 47 further comprising the step of:

replacing, by said supervising program, within said tag table said hash function value of said tag table sent in said previous call-up message by said hash function value of said tag table sent in said current call-up message.

55. The method of claim 47 comprising the further step of:

replacing, by said supervising program, within said tag table said continuation message received in said previous call-up by said continuation message received in said current call-up.

56. The method of claim 46 wherein said call-up to a guardian center occurs each time an operating system or said supervising program are loaded into memory in said user device.

57. The method of claim 47 further comprising the step of:

09706074.110300

measuring, by said supervising program, an elapsed time between a first call-up to a guardian center and a second call-up to a guardian center, by use of one or more event counters.

58. The method of claim 57 wherein said event counters are updated periodically as recorded by a clock.

59. The method of claim 57 further comprising the steps of:

storing by said guardian center, a current time value in said continuation message; and

setting, by said supervising program, an event counter to said current time.

09706474.110300

60. The method of claim 47 further comprising the steps of:

storing user device descriptive values in said tag table;

storing by said supervising program, a plurality of tag tables, said tag tables having said tag table identifier value of said tag table whose hash function values were sent to said guardian center in a plurality of most recent call-ups;

storing, by said guardian center, in said continuation message, a plurality of said hash function values of said tag tables sent in said plurality of said most recent call-ups; and

upon receiving said continuation message, said supervising program, computing said hash function values of said stored plurality of tag tables and further verifying that said hash function values are equal to said corresponding values in said continuation message.

61 The method of claim 60 further comprising the step of:

checking, by said supervising program, whether said user device descriptive values in said tag tables sent in said plurality of most recent call-ups belong to a plurality of user devices.

62. The method of claim 61 wherein the step of checking further comprises the step of:

searching said plurality of tag tables for two successive tag tables including user device descriptive values which differ by more than a specified number of corresponding values.

63. The method of claim 62 wherein the step of checking further comprises the step of:

searching said plurality of tag tables for a first tag table, a second tag table and a third tag table, the second tag table sent in a call-up that occurred later than a call-up in which said first tag table was sent and said third tag table sent in a call-up that occurred later than said call-up in which said second tag table was sent and wherein said user device descriptive values stored in said first tag table and in said second table differ in more than a specified number of corresponding values and said user device descriptive values stored in said first tag table and in said third tag table differ in fewer than a specified number of corresponding values.

64. The method of claim 61 further comprising the steps of:

forwarding, by said supervising program, said result of said verification to said guardian center; and

disabling future call-up messages including said tag table identifier value by said guardian center upon determining that said tag tables sent in said plurality of most recent call-ups belong to a plurality of user devices.

09705074.110300

65. The method of claim 47 wherein said call-up message includes a new randomly chosen value occurring only once.

66. The method of claim 46 wherein said continuation message includes a superfingerprint.

67. The method of claim 66 further comprising the step of:

computing, by said guardian center, a hash function value of a portion of superfingerprints included in continuation messages sent to said supervising program in previous call-ups and in said continuation message;

storing, by said guardian center, said hash function value in said continuation message forwarded to said supervising program;

verifying by said supervising program, that a hash function value of a corresponding portion of said superfingerprints stored on the user device and included in said continuation message is equal to said received hash function value; and

appending, by said supervising program, on said user device said new superfingerprint to said superfingerprints stored on said user device.

09706074.110300

68. The method of claim 46 wherein said call-up message includes said current user device time on said user device.

69. The method of claim 68 wherein the step of verifying further comprises the step of:

verifying, by said guardian center whether said current user device time is within a specified tolerance of a clock time on said guardian center.

70. The method of claim 46 wherein the step of verifying further comprises the step of:

verifying, by said guardian center that a time difference between said arrival of said call-up message and said previous call-up message exceeds a specified minimum.

71. The method of claim 46 wherein the step of verifying further comprises the step of:

verifying by said guardian center that a time difference between said arrival of said call-up message and said previous call-up message is below a specified maximum.

72. The method of claim 47 further comprising the step of:

upon receiving said continuation message, verifying by said supervising program, that said total usage measured across all items in said tag table exceeds said total usage measured across all items in said tag table sent associated with said previous call-up message.

73. A user device comprising:

user device descriptive values; and

a supervising program which records said user device descriptive values.

74. The user device of claim 73 wherein said user device descriptive values include processor-identifying information.

75. The user device of claim 73 wherein said user device descriptive values include non-volatile storage device-identifying information.

76. The user device of claim 73 wherein said user device descriptive values include directory structure identifying information.

77. The user device of claim 73 wherein said user device descriptive values include file identifying information.

78. A software checker comprising:

a superfingerprint, said superfingerprint including data and a computer program;

a guardian center which sends a plurality of superfingerprints for a copy of software to a user device; and

a supervising program executing in said user device which stores the plurality of superfingerprints.

09709474.030600

79. The software checker of claim 78 wherein said superfingerprint further comprises a copy of software name, said copy of software name indicating said copy of software to be checked.

80. The software checker of claim 78 wherein said superfingerprint further comprises a weight, said weight determining said frequency of use of said superfingerprint for checking said copy of software.

81. The software checker of claim 78 wherein said superfingerprint further comprises a list of hash function values of portions of a copy of software.

82. The superfingerprint of claim 81 further comprising a hash function.

83. The software checker of claim 78 further comprising:

a decryption program within said superfingerprint.

84. The software checker of claim 78 wherein said superfingerprint includes a monitoring program which monitors said behavioral characteristics of a copy of software.

85. The software checker of claim 78 wherein said superfingerprint further comprises a public key of a vendor associated with said copy of software.

86. The software checker of claim 78 wherein said guardian center sends said superfingerprint in a digitally signed message to said supervising program.

87. The software copy checker of claim 86 wherein said supervising program verifies said digital signature and stores said superfingerprint if said verification is successful.

88. A method for examining a copy of software used in a user device comprising the steps of:

providing a plurality of superfingerprints, a superfingerprint including a value, a program, a condition, and location information;

executing said program on a portion, said portion specified by said location information, of said contents of a copy of software and computing a value; and

verifying that said pair of said computed value and said included value satisfies said condition.

89. The method of claim 88 further comprising the steps of:

storing a weight in said superfingerprint; and

selecting said superfingerprint to test dependent on said weight.

90. The method of claim 88 further comprising the step of:

09705074 . 110300

providing at least one tag wherein said tag is digitally signed by a vendor; and

verifying that a copy of software used in a user device has an associated tag.

91. The method of claim 90 further comprising the step of:

taking a punitive action upon said successful verification of said condition and said failure of said verification of said copy of software having an associated tag.

92. The method of claim 88 further comprising the further step of:

taking a punitive action upon said successful verification of said condition and an absence of any tag on said user device.

93. The method of claim 90 wherein said associated tag includes said name of said copy of software.

94. The method of claim 90 wherein said associated tag includes a hash function value of a portion of said copy of software.

95. The method of claim 88 wherein said program includes a hash function, said value is a list of hash function values, and the step of verifying of said condition further comprises general-location hash function value checking.

96. The method of claim 88 wherein said program includes a hash function, said value is a list of hash function values, and the step of verifying said condition further comprises same-location hash function value checking.

09705242.110300

97. The method of claim 88 wherein said program monitors behavior of a used copy of software, said value includes a list of actions, and the step of verifying said condition further comprises comparing said monitored behavior against said list.

98. The method of claim 88 wherein said program evaluates intermediate results produced by software, said value includes a list of results, and the step of verifying said condition further comprises comparing said evaluated intermediate results with said list.

99. The method of claim 88 wherein said copy of software is a computer program and said location information specifies a sequence of counts of bytes starting at said beginning of said computer program.

09705074.140300

100. The method of claim 88 wherein said copy of software is a computer program and said value is a list including an instruction.

101. The method of claim 99 wherein no-operation instructions are excluded from said counts.

102. The method of claim 100 wherein location information in said superfingerprints excludes certain portions of instructions.

103. The method of claim 102 wherein said excluded portions of instructions comprise memory locations.

104. The method of claim 102 wherein said excluded portions of instructions comprise register locations.

105. A method for examining a copy of software used in a user device comprising the steps of:

providing a superfingerprint, said superfingerprint including a program using said copy of software;

tracking, by said program, said use of said copy of software; and

recording data related to said use.

106. A method for allowing use of a copy of software on a user device, said copy of software having a tag, comprising the steps of:

obtaining said tag from a tag table in said user device;

computing a hash function value of a portion of said copy of software;

comparing said computed hash function value with a hash function value stored in said tag; and

allowing said use of said copy of software upon successful verification of equality of said values.

107. The method of claim 106 wherein the step of comparing further comprises the step of:

checking that said use of said copy of software is allowed by comparing said use with a usage policy stored in said tag; and

said verification further comprises success of said check.

108. The method of claim 106 wherein the step of comparing further comprises the step of:

comparing a tag table identifier value included in said tag with a tag table identifier value for said tag table.

109. The method of claim 106 wherein the step of allowing further comprises the step of:

09706074.110300

recording usage statistics for said copy of software.

110. The method of claim 106 further comprising the steps of:

checking whether a superfingerprint stored in said user device matches said copy of software;

upon detecting a match, verifying that a vendor name and a public key included in said superfingerprint are equal to a vendor name and a public key included in said tag; and

upon failure of said verification, disallowing said use of said copy of software.

111. The method of claim 106 further comprising the steps of:

checking whether a superfingerprint stored in said user device matches said copy of software;

upon not finding any such match, allowing use of said copy of software.

112. A method for supervising use of software on a user device comprising the steps of:

providing a tag table in said user device, said tag table including a tag table identifier value;

sending, by said user device, a call-up message to a guardian center, said call-up message including said tag table identifier value; and

verifying, by said guardian center that said difference between a time of said call-up message and a time of a last call-up message including said tag table identifier value exceeds a specified minimum value.

113. The method in claim 112 further comprising the steps of:

09706074.110300

upon successful verification by said guardian center, generating a digitally signed continuation message, said digitally signed continuation message including said call-up message;

storing, by said guardian center, said call-up message; and

sending, by said guardian center, said digitally signed continuation message to said user device.

114. The method of claim 112 wherein the step of verifying further comprises the step of:

computing a difference between a time as recorded on said user device included in said call-up message and said time as recorded in said guardian center.

09760434 .011301

115. The method of claim 113 wherein said digitally signed continuation message sent by said guardian center further includes a hash function value of a portion of superfingerprints previously sent by said guardian center in response to a call-up message including said tag table identifier value.

116. The method of claim 113 wherein said continuation message includes a new superfingerprint provided by said guardian center.

117. The method of claim 113 further comprising the step of:

verifying, by said user device said signature of said guardian center and said tag table identifier value included in said continuation message.

118. The method of claim 117, further comprising the steps of:

storing a user device time as recorded on said user device in said call-up message;

storing said user device time in said continuation message; and verifying that said time is earlier than by less than a specified value from said user device time upon receiving said continuation message.

119. The method of claim 115 further comprising the step of:

verifying, by said user device, that said hash function value of said portion of previously sent superfingerprints stored in said user device is equal to said hash function value included in said continuation message.

120. The method of claim 116 wherein said digitally signed continuation message sent by said guardian center further includes a hash function value of a portion of said superfingerprints previously sent by said guardian center in response to a call-up message including said tag table identifier value and a superfingerprint sent in said continuation message.

121. The method of claim 120 further comprising the step of:

verifying, by said user device, that said hash function value of said portion of previously superfingerprints sent by said guardian center and stored in said user device is equal to said hash function value included in said continuation message.

122. The method of claim 116 further comprising the step of:

installing by said user device a new superfingerprint in said tag table.

123. A method for ensuring that a user-specified user device identifier value is present on only one user device comprising the steps of:

sending a message from said user device to a receiver, said message including said device identifier value associated with said user device;

searching, by said receiver, a data structure associated with each possible user device identifier value; and determining by an ID-checking procedure whether said user device identifier value is stored on another user device.

124. The method of claim 123 further comprising the step of:

upon determining that a user device identifier value is on a plurality of user devices, invalidating by said receiver said user device identifier value.

09706074 .110300